CMPSC 465 Spring 2024 — Data Structures & Algorithms — Mehrdad Mahdavi and David Koslicki — HW 9

1. (20 pts.) **Set Cover.**

1. The first selected subset is $\{t,i,m,e,r\}$. As all the letters are uncovered elements, we simply pick the set with the most letters (five) and appears first in the list.
2. The next selected subset is $\{d,o,g\}$ because it has three uncovered elements, which is the most compared to other words and appears first in the list.
3. The third subset we pick is $\{g,o,a,t\}$ because it has one uncovered element, which is the most compared to other words and appears first in the list.
4. The last subset we pick is $\{e,y,e\}$ because it has one uncovered element, which is the most compared to other words. After this, all letters/elements are covered.

2. (20 pts.) **Huffman Encoding.**

(a) It's possible to obtain this sequence. Any f_a, f_b, f_c such that $f_a \geq f_b \geq f_c$ can result in this encoding.

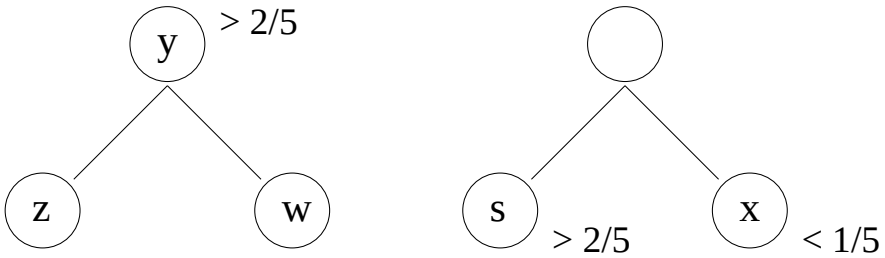
(b) This encoding is not possible since the encoding for a (0), is a prefix of the encoding for b (00).

(c) This encoding also cannot possibly be obtained as it's suboptimal. The encoding for one of the letters in the alphabet could have length 1. For example, the encoding for a could be 0 so that we don't waste space.

3. (20 pts.) **Huffman Properties.**

a Let s be the symbol with the highest frequency (probability) $p(s) > 2/5$ and suppose that it merges with some other symbol during the process of constructing the tree and hence does not correspond to a codeword of length 1. To be merged with some node, the node s and some other node x must be the two with minimum frequencies. This means there was at least one other node y (formed by merging of other nodes), with $p(y) > p(s)$ and $p(y) > p(x)$. Thus, $p(y) > 2/5$ and hence $p(x) < 1/5$.

Now, y must have been formed by merging some two nodes z and w with at least one of them having probability greater than $1/5$ (as they add up to more than 2/5). But this is a contradiction - $p(z)$ and $p(w)$ could not have been the minimum since $p(x) < 1/5$.



b Suppose this is not the case. Let x be a node corresponding to a single character with $p(x) < 1/3$ such that the encoding of x is of length 1. Then x must not merge with any other node till the end. Consider the stage when there are only three leaves - x, y and z left in the tree. At the last stage y, z must merge to form another node so that x still corresponds to a codeword of length 1. But, $p(x)+p(y)+p(z) = 1$ and $p(x) < 1/3$ implies $p(y)+p(z) > 2/3$. Hence, at least one of $p(y)$ or $p(z)$, say $p(z)$, must be

greater than $1/3$. But then these two cannot merge since $p(x)$ and $p(y)$ would be the minimum. This leads to a contradiction.

4. (20 pts.) **Colored Number Line.** First note that an edge should never connect a blue and red point since the edge will be removed. This means that for a sequence RGB, RG should be edge and GB should be an edge. Also note that for RGR and BGB, connecting G to the other two nodes is better than having an edge RR or BB, so subproblems separated by Gs can be formed and solved independently. If there are no Gs or 1 G in the subproblem, the solution is trivially to connect the Rs (Bs) from left to right to their neighbors and the G to the closest R (B). Suppose that each subproblem starts and ends with a G. Since there is no edge RB, consider removing all the Rs (Bs), then the only edges to consider are the edges GG and the edges between neighbors. The common edge of GG can be used by both the R and B cases so the trivial solution used for less Gs is not necessarily optimal. Since the points must be connected, if GG is used, only one edge can be removed from each of the R and B cases, so the minimum subproblem cost will be the minimum of 2 times the length of the segment spanned by the subproblem and 3 times the length of the subproblem minus the longest of the edges that would be connected to a R and the longest of the edges that would be connected to a B. The running time will be linear in the number of points since the longest edges can be found in linear time in the subproblem length so a subproblem can be solved in linear time and the total number of points across subproblems is linear.

5. (20 pts.) **Huffman Efficiency.**

(a) $m\log_2(n) = mk$ bits.

(b) The efficiency is smallest when all characters appear with equal (or near-equal) frequency. In this case, the binary tree that represents the encoding is a complete tree and each encoding takes $\log n = k$ bits. Therefore, encoding the entire file takes mk bits and $E(F) = 1$.

(c) Let F be $x_0, x_1, \ldots, x_{n-2}$ followed by $m-(n-1)$ instances of x_{n-1}. x_{n-1} will be encoded as a single bit, with all of the others being approximately $\log_2(n)+1$ bits (because n is a power of 2, one of the infrequent symbols will be encoded using $\log_2(n)$ bits, but this minor difference will be lost in the big-O notation). This file has efficiency:

$$\frac{m\log_2(n)}{(m-(n-1))\cdot 1+(n-1)\cdot(\log_2(n)+1)} = \frac{m\log_2(n)}{m-(n-1)+(n-1)\log_2(n)+(n-1)}$$

Assuming m is very large, we have the following big-O notation:

$$= \frac{m\log_2(n)}{O(m)+O(n\log n)} = \frac{m\log_2(n)}{O(m)} = O(\log(n))$$

So the efficiency is $O(\log(n)) = O(k)$

6. (0 pts.) **Acknowledgments.**

(a) I did not work in a group.
(b) I did not consult with anyone other than my group members.
(c) I did not consult any non-class materials.

Rubric:

Problem 1, 20 pts

- 3 points: for each correct selected subset.
- 2 points: for correctly identifying the number of uncovered elements in each subset.

Problem 2, 20 pts

(a) 3 points: correct conclusion
3 points: correct set of frequencies

(b) 3 points: correct conclusion
4 points: correct explanation i.e. correctly point out the issue

(c) 3 points: correct conclusion
4 points: correct explanation i.e. correctly point out the issue

Problem 3, 20 pts

(a) 10 pts for right proof

(b) 10 pts for right proof

Problem 4, 20 pts

- 15 points for correct running time
- 2 points for correctness explanation
- 3 points for running time analysis

Problem 5, 20 pts

(a) 6 pts for correct answer.

(b) 6 pts: 3 for $E(f) = 1$, 3 for all frequencies equal.

(c) 8 pts: 4 for correct frequencies, 4 for big-O analysis.